ABSTRACT OF THE DISCLOSURE

The problem of sequentially "squeezing" small fields of data in a larger data path in and out of a memory device can be solved in an algorithmically driven memory tester by defining sub-vectors to represent data in the small field, where a sequence of sub-vectors represents the data that would be represented by a full sized vector if such a full sized vector could be applied to the DUT. A programming construct in the programming language of the algorithmically driven memory tester allows sub-vectors to be defined, as well as an arbitrary mapping that each is to have. The arbitrary mapping is not static, but changes dynamically as different sub-vectors are encountered. Arbitrary dynamic mappings change as sub-vectors are processed, and may include the notion that, during the activity for a sub-vector, this (or these) bit(s) of a vector do not (presently) map to any pin at all of the DUT. The arbitrary dynamic mapping is implemented by a collection of MUX's configured by data stored ahead of time in an SRAM, in accordance with what defining program constructs are encountered by the compiler as it processes the test program. A dynamic reverse mapper, also a collection of MUX's similarly controlled by an SRAM, serves as a de-serializer that assembles a sequence of received sub-vectors into a final received full-sized vector.